DOCUMENT RESUME

ED 427 538 FL 025 721

AUTHOR Cheah, Yin Mee; Cheung, Wing Sum
TITLE What Do You Mean by "User-Friendly"?: Pre-Service Teachers Evaluate Language Software.
PUB DATE 1999-00-00
NOTE 19p.
PUB TYPE Reports - Research (143)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Computer Software; *Computer Software Evaluation; *English (Second Language); Evaluation Criteria; Foreign Countries; Higher Education; *Language Teachers; Preservice Teachers; Second Language Instruction; *Teacher Attitudes; Teacher Education
IDENTIFIERS Singapore

ABSTRACT

A survey of 24 English-as-a-Second-Language teacher trainees in Singapore elicited their top five criteria for evaluating language-learning software and their interpretations and understanding of one criterion, "user-friendliness." It was found that in general, this was the first criterion listed for software evaluation; the others were, in descending order of mention, appropriateness, interest, use of animated graphics, educational value, quality of being fun to use, and relevant content. The most popular definitions for "user-friendly" included: simple to use or having simple instructions; having help easily available; ease of navigation; useful icons; and technical knowledge not required. The potential for ambiguity in use of the term is highlighted. It is noted that a majority of the respondents saw themselves as below-average in computer expertise and were concerned about the amount of time available for using technology in a highly-structured classroom and curriculum, and lack of technical support. Implications for policy and practice are examined briefly. Contains 19 references. (MSE)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED 427 538

What do you mean by "user-friendly"?: Pre-service teachers evaluate language software.

Yin Mee Cheah (PhD)
54, Minbu Road,
#07-03, Minbu Court,
Singapore 308184.
E-mail: yinmee@cyberway.com.sg

Wing Sum Cheung (PhD)
National Institute of Education
469, Bukit Timah Road
Singapore 259756.
cheungws@nievax..nie.ac.sg)

Abstract

In this paper, we report our findings of a study where Singaporean pre-service teachers were asked to list their top five criteria for software evaluation. Specifically, we discuss their interpretations and understandings related to one criterion -- "user-friendly", and show how this term has different meanings for each of them. Central to the paper is the theme of software evaluation. We argue that the process of evaluation cannot be a neutral activity where the main means of evaluation is through the use of a list of established criteria. Instead, there is a need to consider contextual factors which are often sociocultural in nature. Our students' choice of criteria for evaluation is an example of how such factors play a significant role in deciding what matters in evaluating software for their classes.

INTRODUCTION

Singapore is a small island nation of 3.2 million people with just over 35 years of independent history and no natural resources. Singapore's survival as a nation depends on its ability to remain competitive in the world economy. As early as 1985, the government already anticipated that the growth of information technology (IT) would have a direct impact on the world GDP (National IT Plan, p.1). The report further identified that "IT will be a key technology in improving business efficiency and labour productivity, and generating new business" (ibid.). It is therefore inevitable that Singapore would restructure its economy to move upstream into highly skilled, high value-added and technology-intensive industries. In line with this, the

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

PERMISSION TO REPRODUCE AND DISSEMINATE THIS MATERIAL HAS BEEN GRANTED BY
Yin Mee Cheah
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

FL 025721

BEST COPY AVAILABLE

goals of education were similarly reorganised towards developing a more IT-oriented workforce and population.

In April 1997, the Singapore government launched a S$2 billion Information Technology (IT) master plan to introduce the use of computers into schools. Among the objectives of the plan is to " provide a broader base of access to IT among our young so as to achieve a levelling up in learning opportunities. Every child will be able to enhance his learning through an IT-enriched curriculum and school environment" (Master plan for IT in Education, p. 1, 1997). Central to the IT-enhanced curriculum, at the primary level especially, is the use of software that is related to the existing curriculum in English, Maths, Science and Chinese.

Training is provided to teachers to equip them with the necessary skills to deal with the new hardware and software, but the job of selecting and evaluating software remains the task of the teacher. This is because every classroom provides a different context and the children themselves are different in their response to and use of each piece of software. The increasing use of CD-ROM software to supplement the curriculum will mean that software selection and evaluation will be a new and challenging activity for all Singapore teachers.

The National Institute of Education (NIE), the sole teacher-training institution in Singapore, has already embarked on a number of new courses designed to train pre-service teachers to meet the demands of the IT master plan. Among the courses offered is one on the selection, evaluation, and use of software for English language

learning. In this course, student teachers work with educational software and develop ways of evaluating and rating the software used. We found that NIE student teachers, like the teachers described elsewhere in the literature, have a number of different criteria for selecting language software.

In this paper, we would like to begin by discussing the issues related to the selection and evaluation of software before reporting on some of the criteria our pre-service teachers have come up with to guide their own selection. Central to the paper, however, is an investigation into the different interpretations these teachers have for the criteria used for evaluation purposes. We will look specifically at the use of the term "user-friendly" and analyse the many meanings this term has for student teachers. Implications for software evaluation will be discussed in the conclusion.

SOFTWARE EVALUATION - THE ISSUES

Gill, Dick, Reiser, & Zahner (1992) maintained that the most prevalent approach to selecting software is to review evaluations published by software evaluation services. Most of these evaluations are done through the use of a checklist by reviewers who are required to make subjective decisions about the accuracy of the content and the effectiveness of the package for the classroom, as well as its technical strengths and weaknesses. The reliability of such reviews has been questioned by a number of researchers cited in Gill et al. (for e.g. Dudley-Marling & Owston, 1987; Jolicoeur & Berger, 1988; Komoski, 1984; Muller, 1985), but this approach is still used because of its convenience. It is also the fastest way to provide some information about any new piece of software.

This same strategy for software selection is popular in Singapore too, but the problem is magnified many times because many of the reviews and evaluations are done in contexts that are outside Singapore, notably in Western, native-speaking English countries. In fact, English language CD-ROM software produced for language learning is often developed in English-speaking countries for monolingual speakers of the language. Software for speakers of English as a second language (ESL) is not easily available; bilingual software packages with English and the appropriate native languages are even fewer.

Singapore is a multilingual and multiethnic society with a bilingual education policy that requires students to learn English as a first school language and one other official language (Malay, Mandarin or Tamil) as a second language. Since Singaporeans are learning English as the first school language, software in English language is in great demand, and software programs for such skills as reading, vocabulary, grammar are also eagerly sought after particularly after the launch of the IT master plan. Local software production is still in its infancy stage, and there is no choice but to rely on foreign countries to supply the software. Most of the software currently used in schools is imported from the United States.

Software reviews that are available to guide teachers in their selection are almost always not directed at local teachers or local school population. Although there may be generic learning situations and activities that are applicable across situations, these are limited, and reviews (except Singaporean reviews when available) do not point out

the exceptions to Singapore teachers. Because of this, there is a need for teachers themselves or for local reviewers to review the software with the local context in mind. But when this happens, reviewers tend to rely on established checklists of criteria since none have been developed locally. Established lists and criteria are not without their problems, but more of this later.

Next, a number of issues can be identified in relation to the very notion of software evaluation. The first has to do with what the process of evaluation entails. For some, evaluation is a process of description and appraisal of software by teachers or reviewers while for others the process includes pre- and post- testing and observations of student use (Centre for Educational Research and Innovation, 1989). The former notion of evaluation is probably the most popular, but it is the latter process that provides the most useful information for teachers. These two strategies also reflect a formative vs. summative approach to evaluation. In a formative evaluation according to Knussen, Tanner, & Kibby (1991: 14), the concern is with progress towards achieving the goals of an educational innovation, but a summative evaluation is concerned with the effectiveness of a programme upon completion in relation to its stated aims. Summative evaluations are rarely available to the public.

The issue of who the evaluators are is also important. Heller (1991) pointed out that evaluation is often carried out by a wide range of individuals and groups ranging from students, classroom educators, administrators and librarians as well as members of professional associations. These reviews are, as expected, subjective reviews that are much influenced by the individual's knowledge, expertise and interest. A study by

Jolicoeor & Berger (1986) showed that there was little correlation between recommendations given by 2 review services, while Callison and Haycock (1988) found weak correlation between program ratings given by teachers and pupils. Burt (1985, quoted in Heller) and Knussen, Tanner, & Kibby (1991) quoting Lawton (1980) identified teachers as the most appropriate persons to evaluate the software because they are the ones who are going to use it and they know best the context and condition under which the software will be used. However, teachers rarely have the time to do extensive reviews of software before using it in the classrooms. They therefore tend to rely on published reviews as guidelines.

What to evaluate in a software program is yet another issue. Reviews could focus on the content, the technology, the pedagogic presentation or all three. Naturally, the reviewers' strengths and interests play an important part in determining the focus of the review, but this will also mean that the review will only be of use to certain groups. This brings us back to the issue of the criteria to be used in software evaluation. A variety of checklists, questionnaires, and strategies exist for this purpose each with its own list of criteria. While the criteria for technical aspects may be applicable across many contexts (although this also varies with the technology available), the same cannot be said for content and pedagogical presentation.

Where content is concerned, we need to take the subject or the academic discipline into consideration when evaluating the software. For example, software dealing with history or geography requires a different set of evaluation criteria than those dealing with say, Science or language. So far, most checklists ignore such differences in

disciplines. Instead, a generic list of criteria is produced and is used to rate the effectiveness of all software programs. This is not helpful to teachers looking for ways of evaluating discipline-specific software.

Next, the content and instructional design of each software program have to be taken into consideration since what is produced in another cultural context may not always fit in with the educational curriculum or the cultural practices of teaching and learning in another context. For instance, the centralised curriculum in Singapore means that all schools must follow the established syllabus and use prescribed materials. There is often little scope for the inclusion of activities and materials that are not in the syllabus. Software programs produced elsewhere do not always have matching curricula content. In addition, activities that are part of the software program may require pedagogical approaches that are not practised here or are not popular with teachers given the constraint of the curriculum and the class size. This issue of cultural fit is important when considering any material for use in classrooms. Software evaluation procedures neglect this aspect and instead treat the activity as a neutral and cultural-free process.

With these issues in mind, it is clear that selecting a piece of software based on commercial or other published criteria can be problematic because of the different contexts and populations that the software has to be put to use in. What teachers value and need in each context can be very different and is dependent upon the set curriculum, the social and cultural contexts for teaching and learning, the established pedagogical procedures, and also the degree of freedom teachers have to innovate.

With this in mind, we were interested in looking at what our Singaporean pre-service teachers look for in selecting software.

RESEARCH QUESTION

In the class that the first writer conducted on software selection and evaluation, the concern was with the issue of what criteria to use in evaluating software programs for English language classes. Students were exposed to, and played with, a variety of language software used for teaching reading (including skills like phonics and comprehension as well as electronic books), writing, grammar, and vocabulary. As part of our activities, we asked the students to make a list of their top 5 criteria for evaluating software. They were also given a list of 22 criteria from the Bitter and Wighton (1987) study, and asked to select their top 5 criteria from this list. The results of this exercise are reported in Cheung and Cheah (in preparation).

While the results of these two exercises gave us some insight into the students' choices, we also realised that in their ranking of the given criteria and in their own list of criteria, they clearly have different interpretations for each of the terms used. The top criterion that the students listed for software selection was "user-friendly", but we could not tell what they meant by "user-friendly" without further probing. We wanted to know if this term had the same meaning for all of them; if not how did the meaning vary?

Because of this, we decided to ask the students to write down in some detail, what they meant by each of the criteria they have put down in their top 5 list. We compiled

their descriptions to arrive at their interpretations of the criteria. In the rest of this paper, we will focus on one criterion - the most important for the students, and that is "user-friendly". We will discuss the students' interpretations of the term "user-friendly" as applied to the selection and evaluation of language software for teaching and learning English.

USER - FRIENDLY

The term "user-friendly" was originally used to describe software that is easy to install and presents few technical problems for the user. Over the years, however, it has spread to other domains of use. In fact, it is equally, if not more popularly, used in non-computer related fields; for example, we can speak of user-friendly doorknobs, or even user-friendly texts. The many uses of the word in many contexts unfortunately only serve to make its meaning less precise.

Rowe (1993: 282) described software friendliness as possessing " clarity and detail of instruction and the ease with which the software can be used without direct teacher supervision". Heinich, Molenda, Russell & Smaldino (1996:254) described "user-friendly" software as that which "makes the computer transparent in the learning process". In Bitter and Wighton (ibid.), the term "user-friendly" is equated with "easy to use", and in this case, the criterion is derived from a collation of a variety of terms all related to or suggesting "easy to use".

In recent years, advances in technology have resulted in software that is less likely to pose technical problems. Besides, the increase in the quantity and quality of technical support in many institutions has reduced the concern with technical difficulties in

software installation particularly for the end-user. Instead, educational issues in the use of software have become more urgent.

In these definitions of "user-friendly", the focus is on friendliness to the user who in both cases is the student. But, the question of "easy to use for whom?" is a pertinent one where teachers are concerned because in many cases, they themselves are struggling to learn how to use the computer or the software. In such an instance, when asked to rate the criteria for selection of software, it is conceivable that teachers, as in our case, were thinking more of themselves as users and were projecting their own fears and concerns rather than considering their students' needs. This explanation might account for the findings we had with regard to the criteria given by our pre-service students.

THE SAMPLE

There were 24 student teachers in this class. They were in their fourth and final year of study, and would graduate to join Singapore schools as full-fledged teachers in about 4 months from the time of the study. We chose these students for the study because they were more likely to be involved in using computers for teaching because of their youth, their training, and their familiarity with computers. Two of the students were males, and they were Humanities students except for two who specialised in Physical Education. All were expected to teach English language as part of their teaching duties as full-time teachers. All of them have had some experience with computers since they took some core courses in computer literacy in their first two years at the Institute. In the software evaluation course that they took, they had readings, hands-on activities using CD-ROM software as well a field trip to a school which uses

computers for teaching and learning. The software that they looked at was all related to teaching and learning English language. Almost all the software programs were imported from the United States; there were a few local programs. All students had some teaching experience during the attachment periods to schools throughout their four years with the Institute.

FINDINGS

The top criterion from the students was overwhelmingly "user- friendly". The other criteria in rank order were appropriacy (2), interesting(3), animated graphics(3), educational value (4), fun to use (4) relevant subject content(4). A further 91 qualitative responses were elicited from the students to define or to explain their use of the term "user-friendly". The table below shows the most popular explanations for the term "user-friendly".

Criteria	No. of mentions
1. Simple to use, Simple instructions	20
2. Help easily available	19
3. Easy to navigate	10
4. Useful icons	6
5. No technical knowledge require	5

Fig. 1. Students' most popular interpretations of the term "user-friendly"

The most frequent explanation given for the term "user-friendly" was " clear, simple and concise instructions" which received twenty mentions. The twenty explanations varied, and the total was compiled from such explanations as:

features are easy to use

the interactive process has to be enjoyable and easy

clear instructions are given at each step.

The next interpretation was "help easily available". If the software is not so easy to use, students believed that it is still fine if help is easily available. By this, they meant that there should be the following features:

no need to refer to any books or manuals

step-by-step tutorial. no need to flip through thick instruction handbook.

Guided tutorial

help bubbles

help is easily available when needed

best if a menu is provided

The third explanation was that the software should be easy to navigate through if it were "user-friendly". By this, the students meant that they wanted the software to do these things:

The software should allow student to exit or get out of an activity as and when he/she wants to

The software should not restrict; instead it should function according to the students' wishes.

Ease of navigation backwards, forwards and within the programme page.

Useful icons was yet another definition of "user-friendly". By this, the students meant

not too many icons to clutter up the page

icons can be clicked, should be highlighted

icons instead of commands.

Finally, "user-friendly" also meant that the software should not require users to have technical knowledge before they can use the software. The students would also prefer it if the software does not need excessive installation time.

DISCUSSION AND IMPLICATIONS

From this one criterion of "user-friendly", we can see that students had very many interpretations or meanings attached to it. This simply reinforces what we have long known- that different folks have different meanings for the same thing. This is also a case where a word has been so over-used that it has ceased to have one precise meaning. Thus, "user-friendly" holds different meanings for different people, and when this term is used in a checklist without any further qualifications or definitions, we run into the danger of creating ambiguity instead of providing clarification. Although it can be argued from the students' explanations that "user-friendly" seem

to mean that the software should basically be easy-to-use, this notion of easy-to-use can vary from person to person. Certainly what is easy to use for teachers may not be so for younger students.

In looking at the findings, it is also important to know something about the students and the kinds of teaching contexts they were going into. The majority of these students (58.5%) have rated themselves as below-average in their computer expertise (Cheah and Cheung, 1997) although they claimed to be comfortable using the computer for personal and instructional purposes. This lack of confidence in their own expertise could account for their demand to have "user-friendly" software. This is especially when the term "user-friendly" seems to refer to the more technical aspects of using the software although such technical difficulties have, in recent years, been reduced considerably owing to developments in technology.

In addition, pre-service teachers were also concerned with the amount of time available for using technology in a highly structured classroom and curricula setting and a lack of technical support in the classroom. Singaporean classrooms typically have enrolments of 40-44 students with no teacher aide. This could also explain the demand for time-saving features which are translated into user-friendliness. Student teachers could also be influenced by their own recent experiences with certain pieces of software which have caused some frustration, and this partially explains the demand for useful icons and easy navigation.

The reference to useful icons suggest that icons are not "transparent images that all users can read"(Zammit, 1996: 2). Zammit explained that icons are culturally constructed, and the ability to read them depends upon a person's cultural and socio-economic backgrounds. The ability to understand and use icons is part of the information technology skills needed for an IT-based society. This poses yet another unrecognised difficulty to educators and users of software which is developed in other cultural contexts and societies.

This contextual information about the pre-service teachers is to further drive home the point that software selection and evaluation is very much influenced by the situation that the reviewers find themselves in and the contexts with which they are familiar with. These student teachers were very aware of the conditions under which they have to teach in, and their criteria reflected this awareness. Established reviews of software do not contextualise the review process, and this review-in-a-vacuum, so to speak, is of no help to the classroom teacher. Similarly, lists of established criteria without careful explications are also of little use except as a preliminary guide.

Where assistance to the user is concerned, these teachers also expressed little patience with manuals and instruction books. Again this is understandable in the Singapore context where the curriculum is already crowded, and the computer has to compete for time with other curricula programmes (for e.g. critical thinking skills and national education) that have been introduced recently. On- screen help is preferred to instructional manuals. In their own evaluations of individual pieces of software, student teachers also expressed frustration with software programs that provide too

many alternatives or too many features as they viewed these as contributing to distracting students from the task at hand. Pupils, they argued, will be more interested in playing with the different features rather than concentrate on the task at hand. This is specifically in relation to the software *Creative Writer* where a host of features is included to allow students to play around with the format of their text, to add stickers, to change the fonts etc. In short, it appears that pre-service teachers are rather conservative, but this could also be because of their fear of losing control in class. Experienced teachers may not have this concern and may thus be more prepared to experiment more. They may therefore welcome the inclusion of a variety of features in the software program.

CONCLUSION

In this paper, we tried to find out what pre-service teachers meant when they described a piece of software as "user-friendly". While they did have a variety of explanations for the term, easy-to-use was a unanimous choice although these teachers were prepared to select software that is not as easy-to-use provided help is easily available. Technical simplicity is another explanation offered for explaining "user-friendly", together with easy navigation and useful icons.

Although "user-friendly" was the top choice of our students, it is only ranked 5th in the Bitter and Wighton (ibid.) study. Our students' concern with this factor, we believe, reflected their own situation and experience in using software and the computer and in teaching for they were, at best, novices just about to begin their career as teachers. This illustrates our point that software evaluation as an activity is never neutral, and

that evaluators are always influenced by the context that they are going to use the software in or by their perceptions of what the context is like. Singaporean teachers who use established or published software criteria or allow them to be only guided by the evaluations that periodically appear in foreign computer journals may be sadly misled.

As the National IT plan begins to shape up, and as computer software begins to make regular appearances in classrooms, the business of software evaluation must be taken more seriously, and criteria should be developed that reflect the contextual situation in Singapore. Similarly, local software developers would do well to pay attention to teachers' concerns and views about software if they want their materials to be relevant and well-used in the schools.

REFERENCES

1. Bitter, G. & Wighton, D. (1987) The most important criteria used by the Educational Software Evaluation Consortium. Educational Software Preview Guide. *The Computing Teacher*, **14**, (6), 7- 9.
2. Burt, C. (1985) Software in the classroom: A form for teacher use. *The Computing Teacher*, May, **12.**
3. Callison, D., & Haycock, G. (1988) A methodology for student evaluation of educational microcomputer software. *Educational Technology*, **28** (1), 25-32.
4. Centre for Educational Research and Innovation. (1989) *Information Technologies in education. The quest for quality software*. OECD: Paris.
5. Cheah, Y. M., & Cheung, W. S. (1997) Singaporean pre-service students' experiences and confidence in using computer technology. . In Zahran, H., T. Ottmann, & Zaidah, R. (Eds.) *Proceedings of the International Conference on Computers in Education* (ICCE), pp. 710-718.
6. Cheung, W. S. & Cheah, Y. M. (forthcoming) Pre-service teachers' software review criteria.
7. Dudley-Marling, C., & Owston, R.D. (1987) The state of educational software: A criterion-based evaluation. *Educational Technology*, **27**(3), 25-29.

8. Gill, B. J., Dick, W., Reiser, R. A., & Zahner, J. A. (1992) A new model for evaluating instructional software. *Educational Technology*, 39 - 44.
9. Heinich, R., Molenda, M., Russell, J. D., & Smaldino, S. E. (1996) *Instructional Media and Technologies for Learning*. Prentice-Hall: Englewood Cliffs, New Jersey.
10. Heller, R. S. (1991) Evaluating software: A review of the options. *Computers and Education*, **17**, 4, 281-291.
11. Jolicoeur, K., & Berger, D. E. (1986) Do we really know what makes educational software effective? A call for empirical research on effectiveness. *Educational Technology*, **26**, (12), 7-11.
12. Jolicoeur, K., & Berger, D. E. (1988) Implementing educational software and evaluating its academic effectiveness: Part 1. *Educational Technology*, **28**(9), 7-13.
13. Komoski, P. K. (1984) Educational computing: The burden of insuring quality. *Phi Delta Kappan*, **66**, 244-248.
14. Knussen, C., Tanner, G. R., & Kibby, M. R. (1991) An approach to the evaluation of hypermedia. *Computers Education*, **17**, (1), 13 -24.
15. Lawton, D., The politics of curriculum evaluation. (1980) In *Curriculum Evaluation Today; Trends and Strategies* (Edited by Tawney, D.) Schools Council Research Studies, Macmillan, London.
16. Muller, E.W. (1985) Application of experimental and quasi-experimental research designs to educational software evaluation. *Educational Technology*, **25** (10), 27-31.
17. National IT Plan Working Committee. (1985) *National IT Plan. A strategic framework*. Singapore: Author.
18. Rowe, H. A. H. (1993) *Learning with personal computers: Issues, observations and perspectives*. Victoria, Aust: Australian Council for Educational Research. 1993.
19. Zammit, K. (1996) Literacy in a technological age: Children's understandings of CD-ROM icons. Paper presented at the AARE/ERA Conference in Singapore.